MULTIMEDIA UNIVERSITY®

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2017/2018

ECE1026 ALGORITHMS AND DATA STRUCTURES
(All Sections / Groups)

09 MAR 2018
9:00 A.M – 11:00 A.M.
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 8 pages including the cover page with 4 questions only.

2. Attempt all four questions. All questions carry equal marks and the distribution of marks is given in the questions.

3. Please print all your answers in the answer booklet provided.

4. State all assumptions clearly.

Question 1

(a) Given the finite state machine in Fig. Q1-1, determine the following:

(i) The **regular expression** for the finite state machine.

[4 marks]

(ii) The set of all states (Q), set of all input symbols (∑), initial state (q_0), and set of final states (F) for the finite state machine.

[6 marks]

(iii) Any **2** accepted words for the finite state machine.

[2 marks]

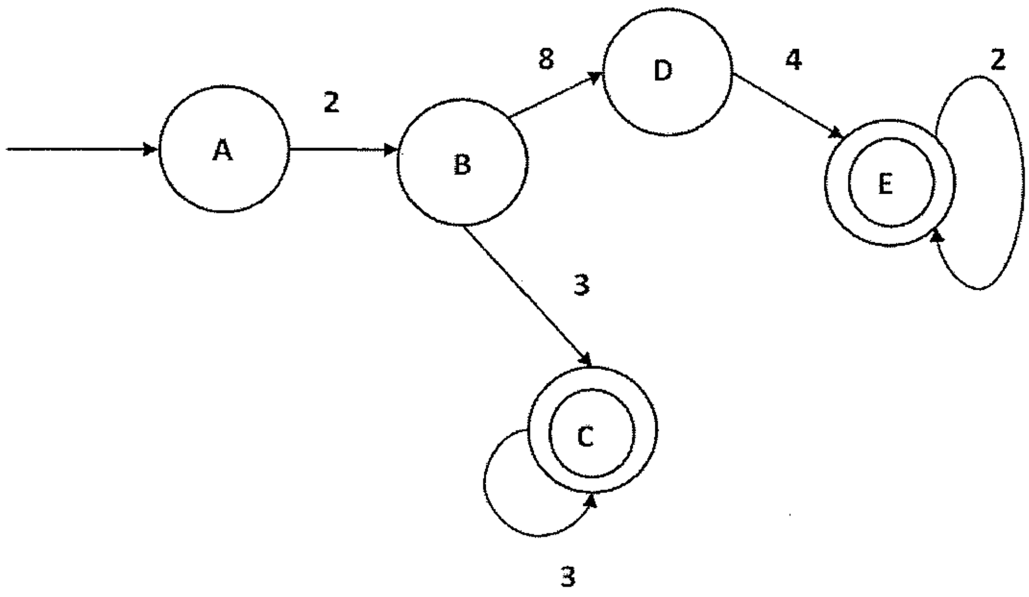


Fig. Q1-1

(b) **Draw** a finite state diagram for regular expression `A*M(B|D+)`.

[5 marks]

Continued ...

Question 1 (continued)

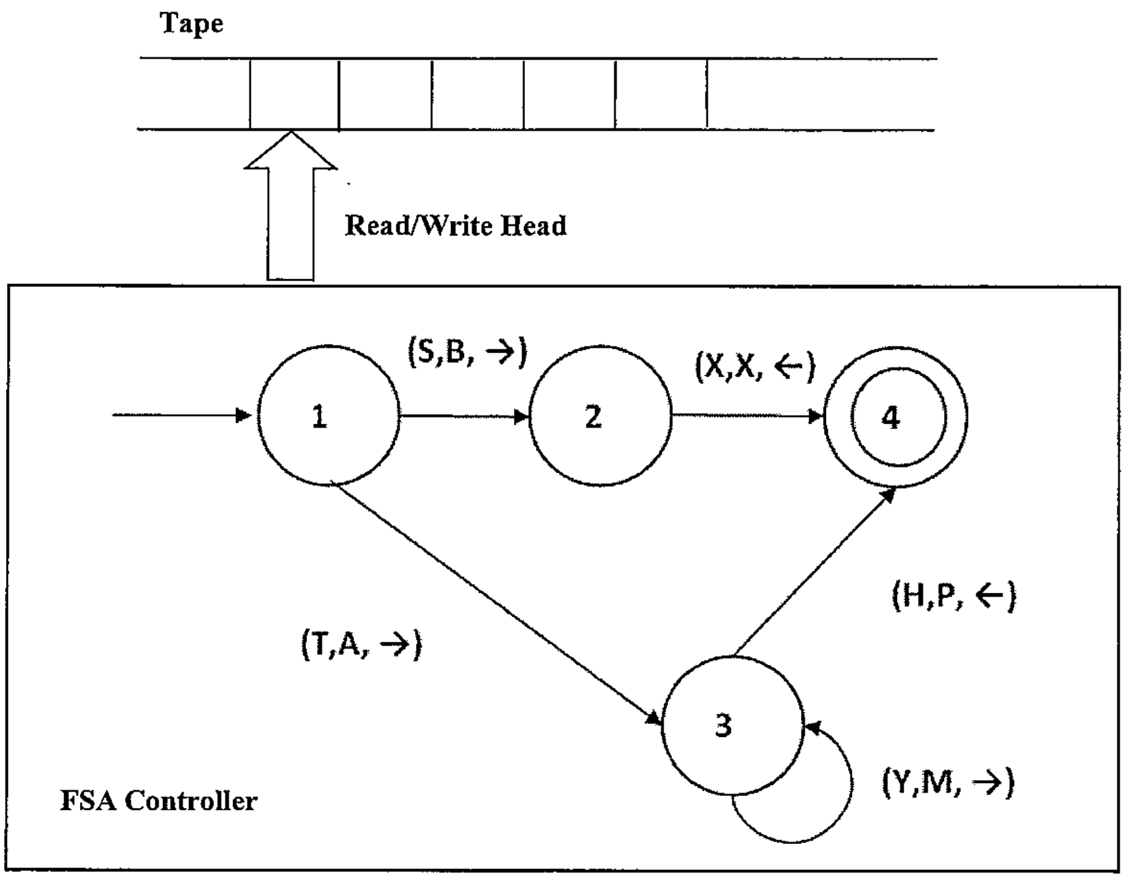
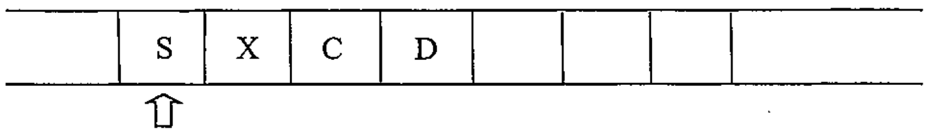


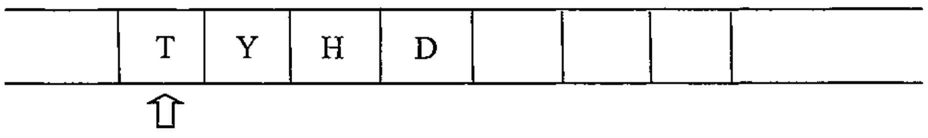
Fig. Q1-2

(c) For each of the following tapes, write down the **contents** of the tape and the **position** of the read/write head at the end of **each** execution cycle until the machine halts for the Turing machine in Fig. Q1-2. Assume that the read/write head is pointing to the left most non blank square in the beginning.

(i) Tape 1



(ii) Tape 2



[8 marks]

Continued ...

Question 2

(a) An incomplete linked list program is given in Fig. Q2-1.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define N 10

//Answer for Part (a)(i)

struct Chess *newChess(char name[ ], int p16, int p17) {
  //Answer for Part (a)(ii)
};

int main() {
  int j = 0;
  char PlayerName[N];
  int p16, p17;
  struct Chess *head, *curr, *highest;

  //Answer for Part (a)(iii)

  //Answer for Part (a)(iv)

  return(0);
}
```

Fig. Q2-1

Table Q2

Chess Palyer Name	Total Points in 2016	Total Points in 2017
Azman	1432	1750
Lisa	1722	1722
Norman	1440	1324

(i) Define a **linked list node** structure named `Chess` that contains a chess player name and the points he or she collected for years 2016 and 2017, see Table Q2.

[3 marks]

(ii) Write the function definition for `newChess` that will create and initialize a new `Chess` node.

[4 marks]

(iii) Write C statements to build a linked list to store data in Table Q2. Use the `newChess` function.

[4 marks]

(iv) Write C statements to **find** and print the player having the highest points in year 2017, as shown below:

```
Highest ranking in 2017 is Azman,
with total points of 1750.
```

[4 marks]

Continued ...

Question 2 (continued)

(b) Fig. Q2-2 shows an incomplete program using graph. Fig. Q2-3 shows the distance graph. Note that the Cyber-Johor link is unidirectional (one way).

```
#include <stdio.h>
#define NUMPLACE 3 //Number of places
#define NOP -1.0   //No path

//These functions are defined elsewhere
int get_origin();
int get_destination();

//Answer for (b)(i)

//Answer for (b)(ii)

int main(void){
   int from, to;

   from = get_origin(); //obtain the index of origin
   to = get_destination(); //obtain the index of destination

   //Answer for (b)(iii)

   return(0);
}
```

Fig. Q2-2

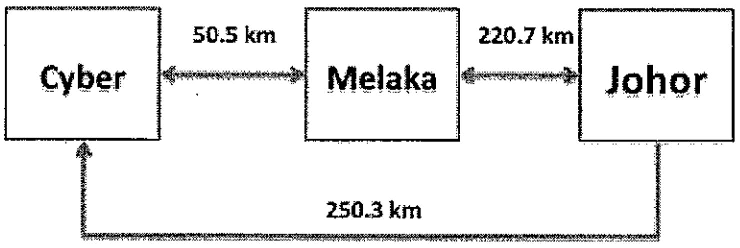


Fig. Q2-3

(i) Declare and define the array of place names.

[2 marks]

(ii) Declare and define the adjacency matrix.

[4 marks]

(iii) Write C statements to print the distance between the origin and the destination, as shown below. If there's no direct path, print "`No direct path`".

```
Cyber - Melaka: 50.5 km
```

[4 marks]

Continued ...

Question 3

(a) Consider the recursive formula:

```
A(n) = 3                 if n = 1;
A(n) = 2A(n-1) + 5       otherwise.
```

(i) Write a direct **recursive** function in C programming language based on the formula above (Assume **n** is an integer greater than 0).

[5 marks]

(ii) Write a **complete** C program to test the recursive function you wrote in part (i). You need to include the recursive function in your answer, showing its position relative to the other parts of the program. Prompt the user for the value of **n** and print the result to the console (screen), as shown in the example below.

```
Enter an integer: 2
A(2) = 11
```

[6 marks]

(iii) Predict the results for **n** = 3, 4 and 5.

[3 marks]

(iv) Describe **two** disadvantages of recursive function.

[4 marks]

(b) Consider the maze in Fig. Q3. '`X`'s are walls, blanks are passages, '`S`' is the start point and '`E`' is the end point. **Sketch** the solution path (including the backtrack path) from point '`S`' to point '`E`' using a **backtracking** algorithm with the following search order: 1) East (right), 2) North (up), 3) West (left), 4) South (down).

[7 marks]

X	X	X	X	X	X	X	X
X				X			X
X				X	X		X
X	X						X
X	X		X	X			X
X	X		S	X			X
X				X		E	X
X	X	X	X	X	X	X	X

Fig. Q3

Continued ...

Question 4

(a) Consider the program listing given in Fig. Q4 on search by hashing. Assume that there is no collision. Write a function definition of each of the following functions: **`initHashtable`** and **`hashSearch`**.

[10 marks]

```
#include <stdio.h>
#include <string.h>
#define TBLSZ 5000

int loadRecords(void);
int hashfunc(char* name); /*the hash function*/
void initHashtable(void);
int hashSearch(char* name);

struct Record {
   char name[200];
   int age;
};
struct Record records[1000]; /*the array of records*/
int hashtable[TBLSZ]; /*the hash table*/
int dbsize; /*number of valid records*/

int main() {
   char name[200];
   int i;

   dbsize = loadRecords(); /*read in the array of records*/
   initHashtable(); /*setup the hash table*/

   printf("Name: ");
   fgets(name, 200, stdin); /*inquire the target's name*/

   i = hashSearch(name); /*hash search*/
   if (i >= 0)
      printf("Age: %d", records[i].age); /*output the target's age*/
   else
      printf("Name not found");

   return 0;
}
```

Fig. Q4

Continued ...

(b) Write an ANSI C function `insertSort` to sort an array of characters in **descending** order (Z to A) using an insertion sort algorithm. The function takes in an array of characters and the array size. The input array is to be replaced by the sorted array.

[6 marks]

(c) Sort by hand the word "`SUPERMAN`" in **ascending** order using the following sorting algorithms. Show all steps involved.

(i) Selection sort

[3 marks]

(ii) Insertion sort

[3 marks]

(iii) Quick sort (the middle element as pivot)

[3 marks]

End of Paper